International Journal of Electrical and Electronics Engineering Research (IJEEER)
ISSN(P): 2250-155X; ISSN(E): 2278-943X
Vol. 5, Issue 1, Feb 2015, 59-74
© TJPRC Pvt. Ltd.

REACTIVE POWER TRANSMIT OPTIMIZATION UTILIZING MULTI-OBJECTIVE EVOLUTINARY ALGORITHM

G KALIDAS BABU[1], RAJESH KUMAR SAMALA[2] & K SWATHI[3]

[1]Associate Professor, Department of Electrical & Electronics Engineering, Nalla Narasimha Reddy Group of Institute, Hyderabad, Telangana, India

[2,3]Assistant Professor, Department of Electrical & Electronics Engineering, Nalla Narasimha Reddy Group of Institute, Hyderabad, Telangana, India

ABSTRACT

This paper presents a Novel Multi-Objective Evolutionary Algorithm used for Optimal Reactive Power (VAR) transmit. The optimal reactive power transmit difficulty is formulated like a non-linear constrained multi-objective optimization difficulty where the real power loss as well as the bus voltage deviations are to be reduced at the same time. A novel technique called NSGA-II as well as Strength Pareto Evolutionary Algorithm (SPEA) based approaches are planned to handle the difficulty as an exact multi objective optimization difficulty. A hierarchical clustering algorithm is compulsory in SPEA to provide the decision maker to maintain controllable pareto-optimal set. Furthermore the results are compared with NSGA-II and found SPEA is slight superior than NSGA-II.

The objective difficulties are formulated by using GA along with different constraints as well as studied utilizing projected techniques. Each and every constraint was satisfied as well as the results obtained for IEEE 30-bus systems are presented, illustrating the effectiveness of the projected approach.

KEYWORDS: Reactive Power and Devices, Constraints, MOEA, SPEA, NSGA-II

INTRODUCTION

Evolutionary Algorithms

Evolutionary Algorithms (EAs) such as evolution strategies and Genetic Algorithms have turn into the technique of preference for optimization difficulties that are too multifarious to be solved utilizing deterministic methods such as linear programming or gradient (Jacobian) techniques. EAs need a little information about the difficulty being solved, and also they are simple to put into practice, strong, and also inherently parallel. To determine a certain optimization difficulty, it is sufficient to need that one is capable to calculate the objective purpose for a specified set of input parameters. Because of their universality, simplicity of operation, and also condition for parallel computing, EAs frequently take fewer time to finish the optimal resolution than gradient techniques. One of the goals of an ideal multi-objective optimization process is to determine as many pareto-optimal resolutions as possible. Since an EA works with a population of resolutions, a population of pareto-optimal resolutions can be captured in one single simulation run of an EA. If we can attain this it will remove recurring utilize of a single objective optimization technique for determining one different pareto-optimal resolution in each run. This will also remove the requirement of any such parameters like weight vectors, ϵ vectors, target vectors and different others.

Some of the other advantages of having evolutionary algorithms are that they need extremely small information about the difficulty being answered, fewer vulnerable to the shape or continuity of the pareto front, simple to put into practice, strong and could be implemented in parallel environment.

The most important motivation for utilizing EAs to resolve multi-objective optimization difficulties is because EAs deal at the same time with a set of possible solutions which permits us to determine some members of the pareto-optimal set in a single run of the algorithm, instead of having to carry out a series of separate runs as in the case of the conventional mathematical programming methods. In addition, EAs are fewer vulnerable to the figure or continuity of the pareto front, whereas these two issues are a real concern for mathematical programming methods.

- **Non-Dominated Sorting GA:** The technique is recommended by Goldberg which is based on some layers of categorizations of the individuals. Before selection is performed, the population is ranked on the basis of non-domination. To continue the diversity of the population, these classified individuals are mutual with their replica fitness standards. Then this cluster of classified individuals is ignored and another layer of non-dominated individuals is considered. The procedure keeps on until all the individuals in the population are classified. A multi-objective optimization difficulty has a number of objective functions which are to be reduced or maximized at the same time. In this study a recent method called "Strength Pareto Evolutionary Algorithm" is utilized to reduce the objective functions.
- **Non-Dominated Sorting GA-II:** This is the modified edition of the NSGA, called NSGA-II, which is further resourceful, utilizes superiority and also a crowed comparison operator that continues diversity without specifying any additional parameters. The NSGA-II doesn't utilize an external memory as in the previous algorithms. Instead its superior mechanism consists of combining the most excellent parameters with the most excellent off spring attained.
- **Strength Pareto Evolutionary Algorithm**: Zitzler and Thiele in 1998 projected a best evolutionary algorithm, which they called the Strength Pareto Evolutionary Algorithm (SPEA). This algorithm starts with a randomly generated population 'P_0' of size 'N' and an unfilled external population 'P_t' with highest capability on 'N_t'. In any generation, the most excellent non-dominated explanations are copied to the external population set 'P_t'. Subsequently the dominated results in the personalized external population are found and removed from the external population. In this manner, previously found elites which are currently dominated by latest elite results obtain removed from the external population. What remains are only the most excellent elite results. If this process is continued over several generations, there is a hazard of the external population being overcrowded along with non-dominated results. In order to control this overcrowding of population above 'N_t', grouping methods is utilized.

Once the latest elites are protected for the subsequent generation, the algorithm then turns to the current population and also utilizes genetic operators to determine a latest population. In this the primary step is to assign a fitness to each solutions in the population, so that a fitness 'S_i' to each member i of the external population first. The strength S_i is proportional to the number N_i of present population members that an external solution i dominates;

$$S_i = n_i / (N+1) \qquad \text{eq. (a)}$$

where N is the number of population in the current population.

Thereafter the fitness of a current population member j is assigned as one more than the sum of the strength values of all external population which weakly dominate j; $F_j = 1 + \sum_{i \in Pt \wedge i \leq j} Si$ eq (b)

The addition of one makes the fitness of any present population member ‘P_0’ to be more than the fitness of any external population member P_t. this technique of fitness assignment suggests that a solution with a smaller fitness is better.

SPEA- Algorithm

Step 1: Find the most excellent non-dominated set of ‘P_0’ and copy these solutions to P_t, where

P_0, is initial population and P_t is the external pareto set with N and N_t respectively.

Step 2: Find the most excellent non-dominated solutions of P_t of the personalized population P_t and remove all dominated solutions.

Step 3: If $|P_t|> N_t$, utilize a grouping method to decrease the size to N_t. Otherwise keep P_t, unchanged. The resulting population is the external population P_t+1 of the subsequent generation.

Step 4: Allocate fitness to every best solution i Є P_{t+1} by utilizing equation, eq (a) then allocate fitness to every population member j Є P_0 by utilizing equation eq (b)

Step 5: Apply a binary tournament selection with these fitness values in a Minimization sense, a cross over and mutation operator to generate the latest population P_{0+1} of size N from the mutual population P_{t+1}+P_0 of size (N_t + N).

Step 3 and step 5 results in the latest external and present population which are then processed in the subsequent generation. This algorithm continues until stopping criterion is met.

NON-DOMINATED SORTING GENETIC ALGORITHM – II (NSGA-II)

NSGA (Non-Dominated Sorting in Genetic Algorithms) is a popular non-domination based genetic algorithm for multi-objective optimization. It is a very effective algorithm but has been generally criticized for its computational comp- lexity, lack of elitism and for choosing the optimal parameter value for sharing parameter σ share. A customized version, NSGA-II was developed, which has a improved sorting algorithm, incorporates elitism and no sharing parameter requires to be chosen a priority.

The modified version of the NSGA, called NSGA-II, which is more resourceful, utilizes elitism and a crowed comparison operator that keeps diversity without specifying any additional parameters. The NSGA-II doesn’t utilize an external memory as in the previous algorithms. Instead its elitist mechanism consists of combining the best parameters with the best off spring obtained.

The primary step of this technique is to sort the population P according to non-domination. Different algorithms can be utilized for this task. It is important to reiterate that any two members from the similar class cannot be said to be superior to one another with respect to all objectives. The total number of classes, represented as ‘ρ’ in the following equation, depends on the population P and the under lying problem.

$P = U^{\rho}_{j=1} P_j$

Once the classification job is over it is clear that all solution in the first set that is, all i Є P_1, belong to the most excellent non-dominated set in the population. The second most excellent solutions in the population are those that belong to the second set or all members of P_2 and so on. Obviously the worst solutions are those belonging to the final set or all members of P_ρ, where ρ is the number of different non-dominated set in the population.

The fitness assignment process starts from the first non-dominated set and consecutively proceeds to dominated sets. Any solution i of the first non-dominated set is allocated fitness equal to $F_i = N$. This exact value of N is utilized for a particular function. Because all solutions in the first non-dominated set are uniformly significant in terms their proximity to the pareto optimal front relative to the present population.

In an NSGA, the sharing function technique is utilized to supply the diversity of the solution. That is, for each solution i in the front P_i, the normalized Euclidian distance d_{ij} from another solution j in the same front is calculated as follows.

$$d_{ij} = \sqrt{\sum_{k=1}^{|P_1|} \left(\frac{x_k^i - x_k^j}{x_k^{\max} - x_k^{\min}}\right)^2}$$

It is significant to note that the sharing distance is calculated with the decision variables, instead of the objective function values as in an MOGA. Once these distances are calculated, they are utilized to calculate a sharing function value utilizing the subsequent equation with α = 2.

$$Sh(d) = \begin{cases} 1-\left(\frac{d}{\sigma_{share}}\right)^{\alpha} \\ 0 \end{cases},$$ If $d \leq \sigma_{share}$, or else

The sharing function obtains a value between 0 and 1, depending on the distance d_{ij}. Any solution i who has a distance better than σ_{share} from the i^{th} solution contributes nothing to the sharing function value. After all |P1| sharing function values are considered, they are added together to compute the niche count nc_i of the i^{th} solution. The niche count, in some sense represents the number of solutions in the neighbourhood of the i^{th} solution. If there is no other solution exists within a radius of σ_{share} from a solution, the niche count for that solution would be one.

The ultimate task is to decrease the fitness of the i^{th} solution by its niche count and get the shared fitness value. $F_j^{/} = F_i / nc_i$. This method of degrading fitness of a solution which is crowded by several solutions helps importance the solutions residing in less crowded region. This process completes the fitness assignment process of all solutions in the first front. In order to continue the second front, we note the minimum shared fitness in the first front and then assign a fitness value slightly lesser than this minimum shared fitness value.

Crowding Distance

Once the non-dominated sort is concluded the crowding distance is assigned. Since the individuals are selected based on rank and crowding distance all the individuals in the population are assigned a crowding distance value. Crowding distance is assigned front wise and comparing the crowding distance between two individuals in different front is meaningless. The crowding distance is considered as follows:

- For each front F_i, n is the number of individuals.
- Initialize the distance to zero for all the individuals i.e. $F_i(d_j) = 0$,

Where j corresponds to the j^{th} individual in front F_i. For each objective function m

- Arrange the individuals in front F_i based on objective m i.e. I = *sort*($F_{i,m}$).
- Assign infinite distance to boundary values for each individual in F_i. i.e., $I(d_1) = \infty$ and $I(d_n) = \infty$
- For k = 2 to (n − 1)

  $I(d_k)=I(d_k)+[I(k+1).m-I(k-1).m]/f_m^{max}-f_m^{min}$

  Where I(k).m is the value of the mth objective function of the kth individual in i.

Selection

Once the individuals are arranged based on non-domination and with crowding distance allocated, the selection is carried out utilizing a crowded comparison operator ($\prec n$). The comparison is carried out as follows based on:

- Non-domination rank prank i.e. individuals in front F_i will have their rank as $p_{rank} = i$.
- Crowding distance $F_i(d_j)$ $p \prec n\, q$ if $- p_{rank} < q_{rank}$
- Or if p and q belong to the same front F_i then $F_i(d_p) > F_i(d_q)$ i.e. the crowding distance should be more. The individuals are selected by utilizing a binary tournament selection with crowd comparison operator.

Genetic Operators

Real-coded GA's uses Simulated Binary Crossover (SBX) operator for crossover and polynomial mutation.

Simulated Binary Crossover

Simulated binary crossover simulates the binary crossover observed in nature and is give as below.

$$c_{1,k} = \frac{1}{2}[(1-\beta_k)p_{1,k} + (1+\beta_k)p_{2,k}] \qquad c_{2,k} = \frac{1}{2}[(1+\beta_k)p_{1,k} + (1-\beta_k)p_{2,k}]$$

where $C_{i,k}$ is the ith child with kth component, $p_{i,k}$ is the selected parent and β_k (≥ 0) is a sample from a random number generated having the density,

$$p^{(\beta)} = 0.5(\eta_c+1)\beta^{\eta_c}, \quad \text{if } 0 \leq \beta \leq 1 \qquad p^{(\beta)} = 0.5(\eta_c+1)\frac{1}{\beta^{\eta_c+2}}, \quad \text{if } \beta > 1$$

This distribution can be obtained from a uniformly sampled random number *u* between (0, 1). η_c is the distribution index for crossover. That is

$$\beta(u) = (2u)^{\frac{1}{(\eta+1)}} \qquad \beta(u) = \frac{1}{[2(1-u)]^{\frac{1}{(\eta+1)}}}$$

Polynomial Mutation

$$c_k = p_k + (p_k^u - p_k^l)\delta_k$$

Where c_k is the child and p_k is the parent with p_k^u being the upper bound on the parent component, p_k^l is the lower bound and δ_k is small variation which is calculated from a polynomial distribution by using

$\delta_k = (2r_k)^{\frac{1}{\eta_m+1}} - 1,$ if $r_k < 0.5$ $\delta_k = 1 - [2(1-r_k)]^{\frac{1}{\eta_m+1}},$ if $r_k \geq 0.5$

Where r_k is a uniformly sampled random number between (0, 1) and η_m is mutation distribution index.

Recombination and Selection

The offspring population is now combined with the current generation population and selection is performed to set the individuals of the next generation. Since all the previous and current best individuals are added in the population, elitism is ensured. Population is now sorted based on non-domination. The new generation is filled by each front subsequently until the population size exceeds the current population size. If by adding all the individuals in front F_j the population exceeds N then individuals in front F_j are selected based on their crowding distance in the descending order until the population size is N. And hence the process repeats to generate the subsequent generations.

PROBLEM FORMULATION

The optimal VAR dispatch problem is to optimize the steady state performance of a power system in terms of one or more objective functions while satisfying several equality and inequality constraints. Generally the problem can be formulated as follows.

Objective Functions

Real Power Loss (P_L): In this case sum of the losses of all branches in the system is to be minimized. This results in the reduction of active power generation in the system. This in turn saves both the generation cost and creates a higher generation reserves. This objective is to minimize the real power loss in transmission lines that can be expressed as;

$$J_1 = P_L = \sum_{k=1}^{nl} g_k [V_i^2 + V_j^2 - 2V_iV_j \cos(\delta_i - \delta_j)]$$

Where nl is the number of transmission lines, g_k is the conductance of the kth line, $V_i\angle\delta_i$ and $V_j\angle\delta_j$ are the voltages at the end buses *i* and *j* of the k_{th} line respectively.

Voltage Deviation (V_D)

This objective is to minimize the deviation in voltage magnitudes at load buses that can be expressed as;

$$VD = \sum_{k=1}^{NL} |V_k - V_k^{ref}|$$

Where NL is the number of load buses; V_k^{ref} is the pre-specified reference value of the voltage magnitude at the k_{th} load bus. V_k^{ref} is usually set to be 1.0 pu.

Problem Constraints

- **Equality Constraints**

These constraints represent typical load flow equations as follow

$$P_{Gi} - P_{Di} - V_i \sum_{j=1}^{NB} V_j [G_{ij} \cos(\delta_i - \delta_j) + B_{ij} \sin(\delta_i - \delta_j)] = 0$$

i = 1,2……….NB

$$Q_{Gi} - Q_{Di} - V_i \sum_{j=1}^{NB} V_j [G_{ij} \sin(\delta_i - \delta_j) - B_{ij} \cos(\delta_i - \delta_j)] = 0$$

i = 1,2……….NB

Where NB is the number of buses, P_G and Q_G are the generator real and reactive power respectively, P_D and Q_D are the load real and reactive power respectively, G_{ij} and B_{ij} are the transfer conductance and susceptance between bus i and j respectively.

- **Inequality Constraints**
  - Generation constraints

Generator voltage V_G and reactive power outputs Q_G are restricted by their lower and upper limits as;

$V_{Gi}^{\min} \le V_{Gi} \le V_{Gi}^{\max}$, i = 1,2,………..NG where NG number of generators

$Q_{Gi}^{\min} \le Q_{Gi} \le Q_{Gi}^{\max}$, i = 1,2,………..NG

- Transformer constraints:

Transformer tap T settings are bounded as;

$T_i^{\min} \le T_i \le T_i^{\max}$, i = 1,2,………..NT where NT is the no. of transformers.

- Switchable VAR sources constraints:

The Switchable VAR compensations Q_C are restricted by their limits as;

$Q_{ci}^{\min} \le Q_{ci} \le Q_{ci}^{\max}$, i = 1,2,…NC, where NC is the number of capacitors.

- Security constraints:

These includes the constraints of voltage at load buses V_L as follows; $V_{Li}^{\min} \le V_{Li} \le V_{Li}^{\max}$, i = 1,2,………..NL where NL is no of Load buses Aggregating the objectives and constraints, the problem can be mathematically formulated as a nonlinear constrained multi-objective optimization problem as follows;

Minimize [P_L(x,u), VD(x,u)] subjected to; g (x, u) = 0 h (x, u) ≤ 0

Where, x is the vector of dependent variable consisting of load bus voltages V_L, and generator reactive power outputs Q_G. There are a set of control variables, which can be modified to satisfy the load flow equations. Hence X can be expressed as;

$$x^T = [V_{L1} \dots V_{L_{NL}}, Q_{G1} \dots Q_{G_{NG}}]$$

u- is the vector of control variables consisting of generator voltages V_G, transformer tap settings T, and shunt VAR compensations Q_C. Hence u can be expressed as; $u^T = [V_{G1} \dots V_{G_{NG}}, T_1 \dots T_{NT}, Q_{c1} \dots Q_{c_{NC}}]$

g-- is the equality constraints.

h-- is the inequality constraints.

A general multi-objective optimization problem consists of a number of objectives to be optimized simultaneously and is associated with a number of equality and inequality constraints. It can be formulated as; Minimize f_i (x), I = 1, 2,……….N_{obj}

Subject to: g_j (x) = 0   j = 1………M, constraints and h_k (x) <= 0

k = 1……...K, constraints

Where f_i is the i_{th} objective functions, x is a decision vector that represents a solution, and N_{obj} is the number of objectives.For a multi-objective optimization problem, any two solutions x^1 and x^2 can have one of two possibilities, one covers or dominates the others or none dominates the other. In a minimization problem, without loss of generality, solution x^1 dominates x^2 if the following conditions are satisfied: $\forall i \in \{1, 2, \ldots\ldots N_{obj}\} : f_i(x^1) \le f_i(x^2)$

$$\exists j \in \{1, 2, \ldots\ldots N_{obj}\} : f_j(x^1) < f_j(x^2)$$

If any of the above condition is violated, the solution x1 does not dominate the solution x2. If solution x1 dominates the solution x2, x1 is called the non-dominated solution. The solutions that are non-dominated within the entire search space are denoted as pareto-optimal and constitute the pareto-optimal set or pareto-optimal front.

MAIN ALGORITHM (SPEA-METHOD)

- Read all system data including bus data, line data, maximum and minimum limits of Generator bus voltages, Load Voltages, Tap Values, Shunt Values, the bus no. at which taps are connected power, etc
- Form Y-bus with present system data
- Generate Initial Population by considering required number of Strings and Population size.
- Decode the strings and evaluate (finding fitness)
- Calculate the actual values of the parameters
- Check the limits of all calculated values and fix to their violated limits
- Run load flow with modified system parameters
- Check all the constraints
- Calculate Objective function ie. P-Loss and Volt deviation as per equation
- Find Nondominated solution from the above Multiobjective function
- Initialize External Pareto Set to Null
- Copy these Nondominated solution to External Pareto Set,
- Check Nondominated Solutions in the External Pareto set and Delete all dominated Solutions from the Pareto set
- Check the size of the Pareto set, if it is greater than pre-defined size, reduce the size by means of Clustering.

- Calculate the Fitness Values of individuals in both Pareto set and the population as follows;
- Assign fitness 'S_i' to each member i of the external population first. The strength S_i is proportion to the number (n_i) of current population that an external population 'i' dominates, $S_i = n_i / (N+1)$
- Thereafter the fitness of a current population member 'j' is assigned as on more than the sum of the strength values of all external population members which weekly dominate j, Fj = 1+ $\sum_{i \in Pt \wedge i \le j} Si$
- Arrange the chromosomes in descending order of their fitness.
- Apply genetic operators
- Uupdate the chromosome matrix.
- Check the convergence conditions, if satisfied go to step 20 else to step no 4.
- Stop and Print the result.
- Stop and print the data

MAIN ALGORITHM (NSGA-II)

In this case also run the Load Flow and get all the necessary data i.e. real power loss and voltage deviation. Take these as the population for the algorithm. In this, real coded GA is employed. The initialized population is sorted based on non-domination. The fast sort algorithm is described as below for each

- For each individual p in main population P do the following
- Initialize *Sp* = *Ø*. This set would contain all the individuals that is being dominated by p.
- Initialize np = 0. This would be the number of individuals that dominate p.
- For each individual q in P

  if p dominated q then add q to the set Sp i.e. *Sp = Sp U {q}*

  else if q dominates p then increment the domination counter

- for p i.e. np = np + 1
- if np = 0 i.e. no individuals dominate p then p belongs to the first front, Set rank of individual p to one i.e p_{rank} = Update the first front set by adding p to front one i.e $F_i = F_i \; U \{p\}$
- This is carried out for all the individuals in main population P.
- Initialize the front counter to one. i = 1.
- Following is carried out while the ith front is nonempty i.e. $Fi \neq Ø$
- Q = Ø. The set for storing the individuals for (i + 1)th front.
    - for each individual p in front *Fi*

  - for each individual q in Sp (Sp is the set of individuals dominated by p)
  - nq = nq−1,decrement the domination count for individual q.
  - if nq = 0 then none of the individuals in the subsequent fronts would dominate q.Hence set qrank = i + 1. Update the set Q with individual q i.e. *Q = Q U q.*
- Increment the front counter by one.
- Now the set Q is the next front and hence *Fi = Q*.
  - Now the crowing distance is calculated as below
- For each front *Fi*, n is the number of individuals.
- Initialize the distance to zero for all the individuals i.e. *Fi(dj) = 0*, where *j* corresponds to the j^{th} individual in front *Fi*.
- For each objective function m
  - Sort the individuals in front F_i based on objective m i.e. I = *sort(Fi,m).*
  - Assign infinite distance to boundary values for each individual in *Fi*. i.e. I(d1) = ∞ and I(dn) = ∞
  - for k = 2 to (n − 1)

$$I(d_k) = I(d_k) + \frac{I(k+1).m - I(k-1).m}{f_m^{\max} - f_m^{\min}}$$

  - The selection is carried out using a crowded comparison operator (≺n). The comparison is carried out as below based on
- Non-domination rank prank i.e. individuals in front *Fi* will have their rank as $p_{rank} = i$.
- Crowding distance *Fi(dj)*
- p ≺n q if $- p_{rank} < q_{rank}$
- Or if p and q belong to the same front *Fi* then *Fi*(dp) > *Fi*(dq) i.e. the crowing distance should be more.The individuals are selected by using a binary tournament selection with crowed-comparison-operator.
  - Apply Polynomial Mutation (refer above theory)
  - Recombination and Selection (refer above theory) The offspring population is now combined with the current generation population and selection is performed to set the individuals of the next generation.
- Repeat the step till it converge.
- Stop and Print the result.

CASE STUDY

In this study, the proposed method was tested on the standard IEEE 30-bus, 6-generator test system to investigate its effectiveness. The single line diagram of the IEEE 30-bus system is given in appendix-1 figure 14. The system has six generators at bus 1, 2, 5, 8, 11, and 13 and four transformers with off-nominal tap ratio in lines 6-9, 6-10, 4-12, and 27-28. The lower and upper level voltage magnitude limits are 0.95 and 1.1pu at all generator buses 2, 5, 8, 11, and 13 respectively and 1.05pu for the remaining buses including slack bus 1. The lower and upper limits of the transformer tapings are 0.9 and 1.1pu respectively. In this case study, Fast decoupled load flow was run as preliminary step. From the load flow we get the system conditions and losses in the system. Also we can find the voltage deviations in the load buses from the reference value. In this case study the reference value of voltage is taken as 1 pu. This real power loss and voltage deviation is taken as the input to the algorithm. First run the load flow with base case and then take the readings (P_{Loss} and VD) for optimization using linear combination of P_{Loss} and VD.

IEEE 30 BUS SYSTEM LINE DIAGRAM

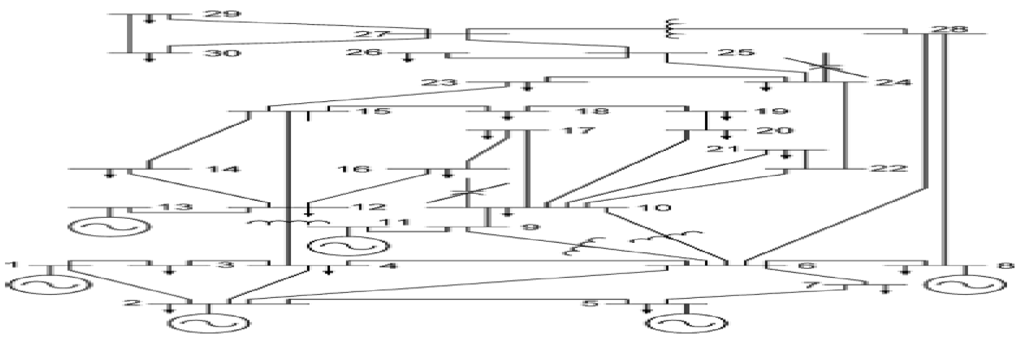

Figure 1

IEEE 30-Bus System: No. of buses: 30 No. of lines: 41

CASE-1 (SPEA): Parameters Selected for Study

No of buses=30 No of lines = 41 Generator buses =6 Tap changing transformers =4 No. of shunts =2

Chromosome Size=250 Total String Length=108 String length for P=07 (each) String Length for V=07 (each)

String Length for Transformer=4 (each) String Length for shunts=.04 (each)

RESULTS

Base Case Voltage Deviation (pu) (VD alone as the objective function)

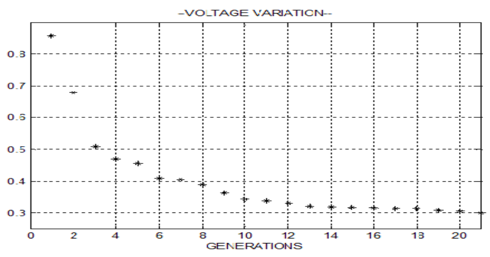

Figure 2: Voltage Deviation Vs Generation

Base Case Power Loss Variation (pu) (PL alone as the objective function)

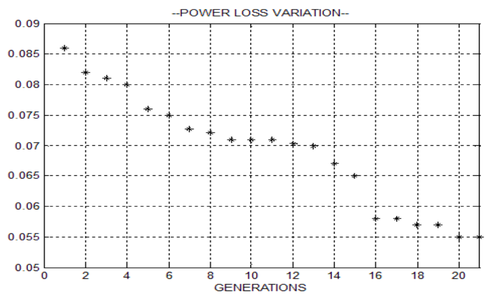

Figure 7: Power Loss Variation Vs Generation Linear Combination of Real Power Loss and Voltage Deviation 30-Bus System

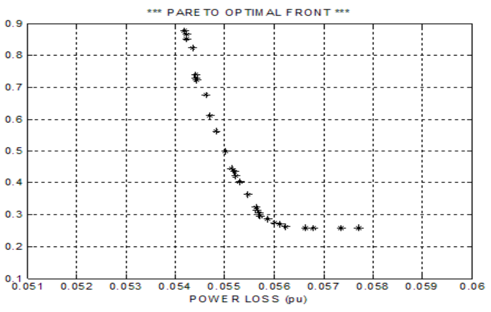

Figure 3: Power Loss Vs Voltage Deviation (250-Population, 150- Generation, 35- Pareto)

Table 1: Result from the Linear Combination of P-Loss and VD

Parameters	Best PL	Best Vd
Power Loss	0.05419	0.0577
Voltage Deviation	0.8783	0.2577
VG-1	1.07	1.05
VG2	1.0424	1.01
VG-5	1.04	1.034
VG-8	1.0287	1.0143
VG-11	1.0632	1.045
VG-13	1.064	1.0523
Tap-1	1.0956	1.0564
Tap-2	0.9872	1.00
Tap-3	1.08	0.976
Tap-4	1.002	0.9671
Shunt	C1= 0.01, C2=0.05	C1=0.04, C2=0.02

No. of Non-dominated Solutions in Single Run case (Linear combination)

POWER LOSS (pu)

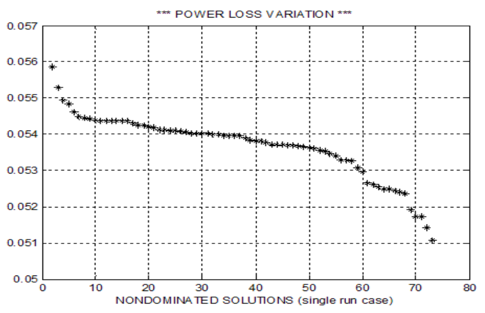

Figure 4: Convergence of Power Loss in Single Run Case

Secondly the problem has been applied to the proposed approach (SPEA) with the above mentioned parameters (i.e. PLoss and VD). Where objective functions, power loss and Voltage deviation on the Load buses are considered simultaneously for the optimization purpose. The problem was run with SPEA and the diversity of the pareto-optimal set over the trade-off surface was found. The result obtained by the proposed method is shown in table 2 for different conditions. The result is also compared with linear combination method and showing that, the proposed method have better diversity characteristics and well distributed over the entire trade-off surface.

Pareto Optimal front in 30 Run case (SPEA)

Table 2: Results Shows the Best Values for 250-pop, 150 Generation

Parameters	Best PL	Best Vd
Power Loss	0.051724	0.05529
Voltage Deviation	0.88143	0.242
VG-1	1.05	1.035
VG2	1.044	1.0386
VG-5	1.0354	1.0254
VG-8	1.03456	0.9938
VG-11	1.0646	1.0268
VG-13	1.09256	1.0445
Tap-1	1.0998	1.055
Tap-2	0.95	0.8823
Tap-3	0.9625	0.9754
Tap-4	0.95	0.9243
Shunt	C1=0.04, C2=0.05	C1=0.03, C2=0.01

Voltage Deviation (Pu)

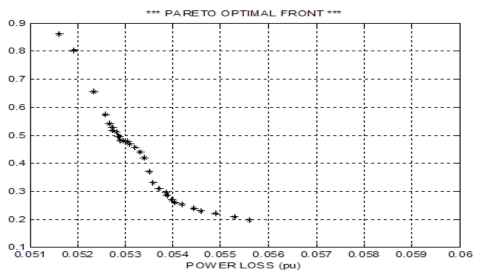

Figure 10: Power Loss Vs Voltage Deviation (250-Population, 150- Generation, 35-Pareto)

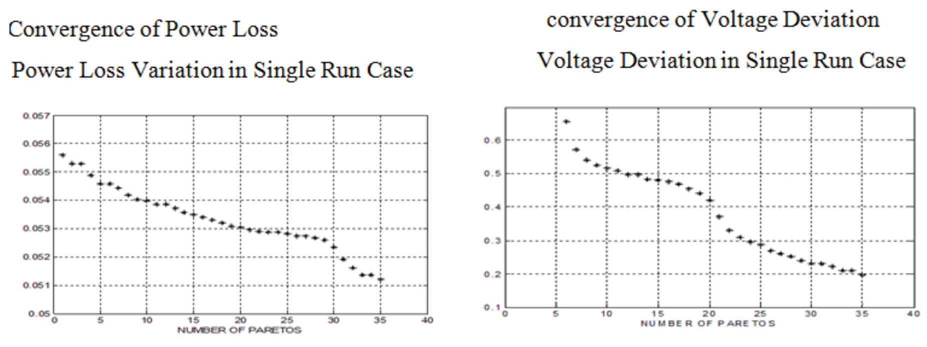

Figure 5

CASE-2 (NSGA-II) Parameters Selected for Study

No of buses =30 No of lines =.41Generator buses =6Tap changing transformers =4 No. of shunts =2

Chromosome Size=200 Generation=500 String length for V=06 (real coded) String length for Transformer = 04(each), 4*4 String Length for shunts=04 (each), 2*4

In this method simulation was done with unadjusted fast decoupled load flow to get the power loss and voltage deviation. Take this objective function as input to the proposed method and run NSGA-II algorithm for optimal dispatch of reactive power.

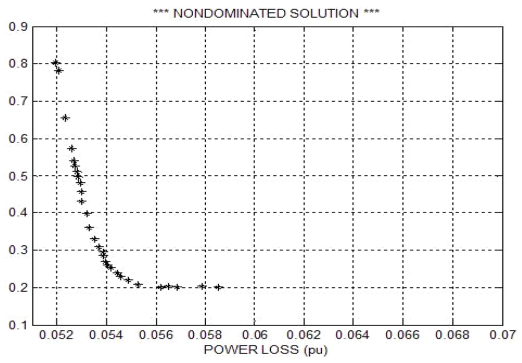

Figure 6: Non Dominated Solutions NSGA-II Voltage Deviation (pu) Voltage Deviation Vs Power Loss

Table 3: Results Shows the Best Values for 200-Pop, 500 Generation (The Best Result from 30 Separate Runs Case)

Parameters	Best PL	Best Vd
Power Loss	0.05209	0.0585
Voltage Deviation	0.782	0.2021
VG-1	1.06	1.05
VG2	1.042	1.1
VG-5	1.03	1.041
VG-8	1.027	1.026
VG-11	1.072	1.05
VG-13	1.084	1.062
Tap-1	1.09	1.05
Tap-2	0.9572	1.00
Tap-3	1.09	0.98
Tap-4	1.01	0.97
Shunt	C1= 0.03, C2=0.05	C1=0.04, C2=0.01

Table: 4 Comparisons of Results Obtained by Different Methods

			Combination					
	Best PL	Best Vd	Best PL	Best Vd	Best PL	Best Vd	Best PL	Best Vd
Power Loss	0.05598	0.0872	0.05419	0.0577	0.051724	0.05529	0.05209	0.0585
VD	0.889	0.301	0.8783	0.2577	0.88143	0.242	0.782	0.2021
VG-1	1.08	1.04	1.07	1.05	1.05	1.035	1.06	1.05
VG2	1.076	1.04	1.0424	1.01	1.044	1.0386	1.042	1.1
VG-5	1.052	1.038	1.04	1.034	1.0354	1.0254	1.03	1.041
VG-8	1.043	1.03	1.0287	1.0143	1.03456	0.9938	1.027	1.026
VG-11	1.071	1.042	1.0632	1.045	1.0646	1.0268	1.072	1.05
VG-13	1.061	1.052	1.064	1.0523	1.09256	1.0445	1.084	1.062
Tap-1	1.074	1.062	1.0956	1.0564	1.0998	1.055	1.09	1.05
Tap-2	1.045	1.05	0.9872	1.00	0.95	0.9823	0.9572	1.00
Tap-3	0.9821	1.025	1.08	0.976	0.9625	0.9754	1.09	0.98
Tap-4	1.0	0.987	1.002	0.9671	0.95	0.9243	1.01	0.97
Shunt	C1=0.04 C2=0.03	C1=0.03 C2=0.05	C1=0.01 C2=0.05	C1=0.04 C2=0.02	C1=0.04 C2=0.05	C1=0.03 C2=0.01	C1=0.03 C2=0.05	C1=0.04 C2=0.01

- **Base Case Voltage Deviation (VD alone as the objective function)**

Voltage Deviation (pu)

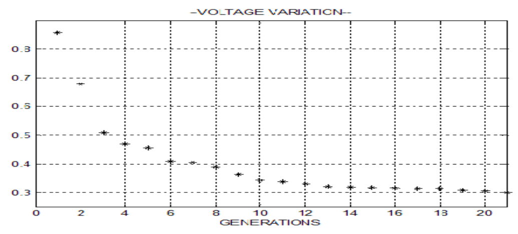

Figure: 6 Voltage Deviation Vs Generation

CONCLUSIONS

- In this paper, a Genetic Algorithm based approach has been presented and applied to multi-objective VAR dispatch optimization problem. The problem has been formulated as multi-objective optimization problem with competing real power loss and bus Voltage deviation.
- A new technique called NSGA-II and SPEA methods are used the study. In SPEA a hierarchical clustering technique is also applied to keep manageable Pareto-optimal set without destroying the characteristics of the trade-off front.
- The result shows that the proposed methods are efficient for solving multi-objective VAR dispatch problems where multiple solutions can be found in one simulation run. Since the proposed approach does not impose any limitations on the number of objective functions, its extension to include more objective is a straightforward process.
- Also this method is useful for nonlinear problems with more objective functions. The result shows that the SPEA is little efficient than NSGA-II for solving multi-objective VAR dispatch problems. But in terms of simulation time, real coded NSGA-II is far better than SPEA.

REFERENCES

1. Mansour MO, Abdel-Rahman TM. Non-linear VAR optimization using decomposition and Coordination. IEEE Trans Power Apparat Syst 1984; PAS-103(2):246–55.
2. Iba K. Reactive power optimization by genetic algorithm. IEEE Trans Power Syst 1994; 9(2):685–92.
3. Hsaio YT, Chaing HD, Liu CC, Chen YL. A computer package for optimal multi-objective VAR planning in large scale power systems. IEEE Trans Power Syst 1994; 9(2):668–76.